Extensões ao Modelo RBAC de Restrições para suportar Obrigações do $UCON_{ABC}$

Edemilson S. Silva[1], Altair O. Santin[1], Edgard Jamhour[1], Carlos A. Maziero[1], Emir Toktar[2]

[1]Programa de Pós-Graduação em Informática (PPGIa) – Centro de Ciências Exatas e de Tecnologia – Pontifícia Universidade Católica do Paraná (PUCPR)
Rua Imaculada Conceição, 1155 – 80.215-901 – Curitiba – PR – Brasil

[2]University of Paris VI, LIP6 Lab. 8,
Rue du Capitaine Scott, 75015 – Paris - France

{edemilson, santin, jamhour, maziero}@ppgia.pucpr.br, emir.toktar@etu.upmc.fr

Abstract. *This work presents a proposal of extension to the model of Role-Based Access control (RBAC) to support activities that demands mutability in their authorization attributes in runtime. Such activities cannot be subdivided in a set of subtasks executed sequentially and nor they can be accomplished by a single role. The approach presented allows the creation of quorum role, which can only be activated in a session with endorsement of a quorum of other roles. A prototype illustrates the application of proposal in a network management scenario. In the illustrative scenario, a previously defined set of roles, by endorsement, activates a quorum role to perform a management task without the participation of the network administrator role.*

Resumo. *Este artigo apresenta uma proposta de extensão ao modelo Role-Based Access Control (RBAC) para suportar atividades que exigem mudanças nos atributos de autorização em tempo de execução. Tais atividades não podem ser divididas em subtarefas a serem executadas seqüencialmente e nem podem ser realizadas por um único papel. A abordagem apresentada permite a criação de papéis quorum, que só podem ser ativados numa sessão com o endosso de um quorum de outros papéis. Um protótipo mostra a viabilidade da proposta em um cenário de gerenciamento de redes. No caso, um conjunto pré-definido de papéis ativa o papel quorum para executar uma configuração na rede sem a participação do administrador da mesma.*

1. Introdução

O controle de acesso baseado em papéis (RBAC) agregou as melhores características dos modelos clássicos, discricionário e obrigatório, no sentido de ser flexível como o primeiro e centralizado como o segundo, o que é difícil de conseguir num mesmo modelo. Porém, o modelo de restrições do RBAC está baseado principalmente na separação de tarefas (Separation of Duty - SoD), suportando o princípio do mínimo privilégio [Salter e Schroeder, 1975]. Assim, a dinâmica do modelo é fortemente influenciada pela separação de tarefas. Outros tipos de restrições foram considerados e

discutidos pelos estudiosos da área, mas não foram adotadas no modelo de referência [Ferraiolo e outros, 2001].

As restrições de separação de tarefas combinadas com o princípio do mínimo privilégio se aplicam bem a cenários onde uma tarefa pode ser dividida em subtarefas executadas em seqüência. Neste modelo não é possível a especificação de políticas em ambientes onde é necessária a autorização (endosso) de um conjunto mínimo de principais[1] (quorum) para a realização de uma tarefa [Shoup, 2000]. Tarefas com este tipo de necessidade geralmente estão associadas a ações em ambiente heterogêneo ou multidisciplinar. Neste caso, os principais devem concordar em se unir para realizar uma atividade que cada um individualmente não teria autorização para efetivar. O quorum, neste contexto, não tem o sentido de tolerância a intrusão ou faltas, mas de exigir um número mínimo (significativo) de principais para obter a autoridade necessária a efetivação de uma atividade.

Os mecanismos de controle de acesso clássicos geralmente verificam a autorização somente na inicialização de uma sessão (logo após o processo de autenticação). Porém, se a política de autorização for mudada durante a sessão os requisitos de autorização podem ser violados. No RBAC as políticas são verificadas quando um papel é ativado, mesmo se a ativação for durante uma sessão. Em contrapartida, quando um acesso é negado por falta de autorização, o mecanismo de controle de acesso não indica qual papel deveria estar ativo para obter tal direito.

O modelo de controle de uso $UCON_{ABC}$ [Park e Sandhu, 2004] é caracterizado por suportar a alteração de atributos (mutabilidade de atributos) do sujeito ou do objeto durante uma sessão. Logo, se houver alteração de atributos durante a sessão, tal modificação será considerada assim que a mesma ocorrer. Porém, um mecanismo que implementa esta característica deve monitorar continuamente os atributos do sujeito e do objeto.

Este trabalho baseia-se na mutabilidade de atributos do $UCON_{ABC}$ para propor a extensão do modelo de restrições do RBAC. Esta extensão permite a ativação de múltiplos papéis (quorum) em uma mesma sessão, com o objetivo de endossar a ação de um papel numa tarefa que um único papel não poderia realizar sozinho.

Um exemplo de cenário envolvendo uma companhia com uma conexão interrompida com a Internet (um roteador BGP fora de operação, por exemplo) e a indisponibilidade do administrador da rede mostra a viabilidade da proposta. Neste cenário, o acesso à Internet precisa ser restaurado urgentemente, mas ninguém da companhia nem os técnicos do fabricante do equipamento possuem autorização para acessar a roteador.

O restante do trabalho está estruturado da seguinte maneira: A seção 2 descreve o modelo de referência do RBAC. A seção 3 apresenta a proposta. A seção 4 mostra um exemplo de aplicação da proposta. A seção 5 considera os aspectos de implementação do protótipo. A seção 6 apresenta os trabalhos relacionados. Finalmente, a seção 7 apresenta as conclusões e indica desenvolvimentos futuros.

[1] São entidades ativas, sujeitos, usuários ou processos.

2. Controle de Acesso Baseado em Papéis (Consensus Model)

O conceito de controle de acesso baseado em papéis (RBAC - *Role-Based Access Control*) surgiu na década de 70. Uma definição mais formal do tema foi instituída pelo NIST (*National Institute of Standards and Technology*) e pelo grupo liderado por Sandhu, definindo o modelo unificado RBAC-NIST [Sandhu e outros, 2000]. Trabalhos subseqüentes definiram o Modelo de Consenso NIST RBAC (*Consensus Model*), padrão definido com base em quatro componentes: RBAC Básico, RBAC Hierárquico, Separação de Tarefas Estáticas e Separação de Tarefas Dinâmicas. Esta organização modular permite aos desenvolvedores implementações parciais do RBAC em seus produtos (definido no RBAC Standard ANSI INCITS 359-2004).

O modelo RBAC Básico inclui cinco elementos básicos: usuários (*Users* – U), papéis (*Roles* – R), objetos (*Objects* – OBS), operações (*Operations* – OPS) e permissões (*Permissions* – PRMS). As relações entre os elementos são mostradas na Figura 1.

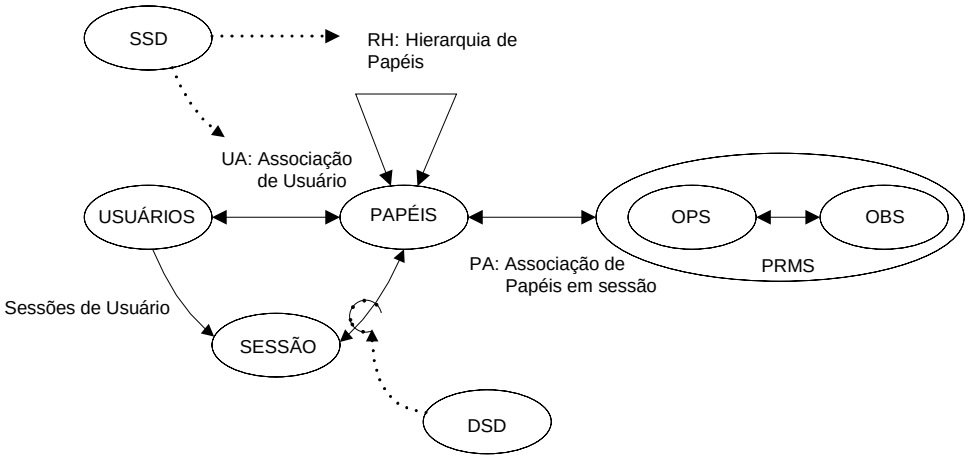

Figura 1. Modelo de consenso NIST RBAC [Sandhu e outros, 2000]

A idéia principal do modelo RBAC é associar as permissões aos papéis e não aos usuários (principais). A Associação de Usuário (*User Assignment* – UA) é um relacionamento "muitos para muitos" (um usuário pode ser associado a um ou mais papéis e um papel pode ser associado a um ou mais usuários). No RBAC é assumido que um papel deve ser ativado numa sessão para que seja possível usar os direitos associados ao mesmo. Uma sessão é associada a um único usuário e cada usuário é associado a uma ou mais sessões. A Associação de Permissões (*Permission Assignment* – PA) é também uma relação "muitos para muitos", entre permissões e papéis. Uma permissão é uma aprovação para realizar uma operação (por exemplo, leitura, escrita etc.) em um ou mais objetos protegidos pelo RBAC (por exemplo, um arquivo, um diretório, uma aplicação etc.).

O modelo NIST RBAC adota o conceito de separação de tarefas (*separation of duty - SoD*). Este conceito consiste em dividir tarefas que podem apresentar conflitos de interesses [Brewer e Nash, 1989] em várias subtarefas executadas por diferentes principais. Este artifício reduz os privilégios associados a usuários individuais (princípio do mínimo privilégio). A separação de tarefas foi discutida em detalhes em [Simon e Zurko, 1997]. Dos tipos de restrições já discutidas por vários pesquisadores da área, o

modelo NIST RBAC suporta apenas a separação de tarefas estática e dinâmica [Ferraiolo e Kuhn, 1992]; este tipo de restrição também está presente de algum modo em outros modelos [Park e Sandhu, 2004] [Nyachama e Osborn, 1999]. O modelo de Separação Estática de Tarefas (*Static Separation of Duty* – SSD) introduz restrições a Associação de Usuário, excluindo a possibilidade de usuários assumirem papéis conflitantes. O modelo RBAC SSD é aplicado a um conjunto de papéis (dois ou mais papéis) e restrições de cardinalidade (alguns papéis podem ter uma limitação no número de usuários associados [Sandhu, 1998]). Por exemplo, para obrigar um usuário a assumir o papel 'r1' ou 'r2' deve ser definido um conjunto {r1, r2} com cardinalidade igual a 1 (o usuário pode exclusivamente assumir um papel do conjunto).

A Separação Dinâmica de Tarefas (*Dynamic Separation of Duty* - DSD) introduz restrições nos papéis que o usuário pode ativar durante uma sessão. A estratégia para imposição de restrições na ativação de papéis é similar à SSD, usando um conjunto de papéis e cardinalidade maior que 1. Note que SSD impõe restrições gerais a quaisquer papéis que um usuário pode assumir, enquanto DSD impõe restrições aos papéis que o usuário pode ativar simultaneamante durante a sessão.

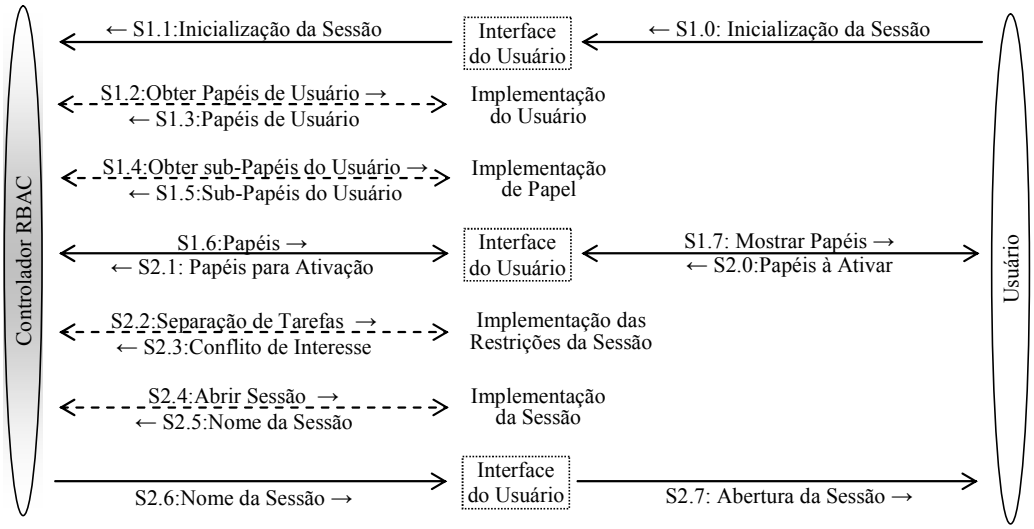

Figura 2. Eventos para a ativação de Sessão

A Figura 2 mostra uma visão geral da dinâmica de estabelecimento de uma sessão RBAC, aplicando restrições. Um usuário inicializa a sessão (evento S1.0) acionando o controlador RBAC através da interface do usuário. No evento S1.2 são recuperados os papéis associados ao usuário e em S1.4 são recuperados seus sub-papéis. Os papéis que o usuário pode ativar na sessão são mostrados ao mesmo no evento S1.7. As restrições de ativação são recuperadas no evento S2.2. O controlador RBAC verifica se o usuário possui autorização para ativar os papéis que escolheu no evento S2.0 considerando as restrições de sessão. Se todos os passos comentados anteriormente ocorrerem com sucesso, os papéis selecionados são ativados e a sessão é aberta para o usuário no evento S2.7.

Em [Park e Sandhu, 2004] foi introduzido o modelo de controle de uso $UCON_{ABC}$. O $UCON_{ABC}$ está baseado em três tipos de restrições: *Autorizações (A), oBrigações (B)* e *Condições (C)*. As autorizações impõem restrições referentes aos atributos de sujeito e objeto que são consideradas no controle de acesso. Obrigações são

ações que os principais devem executar no sentido do controle de uso. As condições são restrições baseadas em variáveis de ambiente ou sistema. Para definir os modelos que compõem a família $UCON_{ABC}$, Park e Sandhu se basearam em três circunstâncias onde as restrições podem ser aplicadas: antes (*pre*), durante (*on*-going) e depois (*post*) de um evento; um evento pode ser uma inicialização de sessão, o acesso a um objeto, o uso de um objeto etc. Note que as restrições *post* aplicadas em um evento e_i influenciarão somente o evento subseqüente, e_{i+1}. Neste artigo, somente as restrições *pre-oBrigações* (preB) e *on-oBrigações* (onB) são relevantes. As restrições de obrigações devem ser consideradas na inicialização da sessão (preB) e durante a sessão (onB).

As restrições introduzidas pelo $UCON_{ABC}$ permitem considerar cenários nos quais os atributos dos principais e dos objetos são modificados dinamicamente, baseados nas ações realizadas pelos principais. Um conceito muito importante proposto no modelo $UCON_{ABC}$ diz respeito a *mutabilidade de atributo*s de sujeitos e objetos [Jaehong e outros, 2004]. Os atributos são dinamicamente modificados como resultado de ações realizadas pelos principais, ou seja, dependem das ações do principal durante a sessão. Atributos de privilégio (A) podem ser alterados durante ou depois de uma sessão. A próxima seção mostra uma proposta de extensão do modelo de restrições do NIST RBAC para suportar as restrições preB e onB definidas no modelo $UCON_{ABC}$.

3. Proposta de Extensão ao Modelo RBAC com Restrições

Este trabalho considera cenários nos quais a aplicação do príncipio do mínimo privilégio (isto é, dividir uma tarefa em subtarefas executadas seqüencialmente) não é possível. Em vez disso, assume-se que um quorum predefinido de papéis autoriza temporariamente um terceiro a realizar uma tarefa que individualmente o mesmo não conseguiria efetivar. Neste caso, os atributos do terceiro (principal) são alterados durante a sessão para refletir a autorização que o endosso do quorum lhe conferiu.

Como exposto na seção 2, o modelo NIST RBAC não se aplica a cenários nos quais atributos de principais podem ser modificados dinamicamente (durante a sessão). Assim, está sendo proposta a extensão do modelo NIST RBAC para suportar as restrições das obrigações (*obligations*) do $UCON_{ABC}$, no caso de controle de acesso baseado em quorum. A extensão proposta define dois tipos de papéis: simples e quorum. Papéis simples são definidos como no modelo original do RBAC, onde um principal pode ativar qualquer papel associado a si desde que atenda as restrições da separação de tarefas. Os papéis quorum só podem ser ativados com o endosso de um quorum composto por um conjunto pré-definido de papéis simples, ou seja, um papel quorum não pode ser ativado por um papel simples.

Os principais que compõem um quorum ativam seus papéis somente para endossar a ativação do papel quorum e não para exercer os privilégios associados aos seus papéis simples. A participação de um principal na ativação de um papel quorum não impede a utilização dos papéis associados a si em uma sessão normal (sem quorum) do usuário. Em outras palavras, qualquer usuário participante da ativação de papel quorum numa determinada sessão pode também ativar esse papel simples em sua própria sessão, quando necessário.

As restrições para ativação de um papel quorum em uma sessão são divididas em restrições pre-oBrigações (preB) e onB (on-oBrigações). Restrições preB definem quais

papéis simples são exigidos para ativar um papel quorum no estabelecimento de uma sessão. Quando o principal já está em uma sessão as preB se tornam onB (obrigações de tempo de execução) – indicando quais papéis simples são necessários para a ativação do papel quorum durante a sessão. As condições definem restrições de ambiente tais como data/hora, endereço de origem/destino etc.

O princípio do mínimo privilégio, SSD e a associação de um papel quorum a principais são interpretados como condições do modelo $UCON_{ABC}$, enquanto que o quorum de principais e DSD denotam obrigações. O mínimo privilégio e a separação de tarefas originários do RBAC não são modificados na proposta. Entretanto, se um principal não possui privilégio suficiente para executar uma tarefa, o mesmo pode receber privilégios adicionais na sua sessão para fazê-lo, desde que tenha previamente associado a si o respectivo papel quorum. Em geral, quanto mais crítica é a tarefa (isto é, quanto mais responsabilidade ou autoridade a mesma requer) maior deve ser o quorum de papéis para efetivá-la.

Um exemplo típico de endosso acontece em agências bancárias onde um caixa pode pagar cheques de até R$ 5.000,00, por exemplo, sem autorização do seu supervisor. Qualquer cheque de valor acima deste valor precisa do endosso (autorização) do supervisor de caixas. A Figura 3 mostra como a proposta de extensão seria aplicada no cenário exemplo.

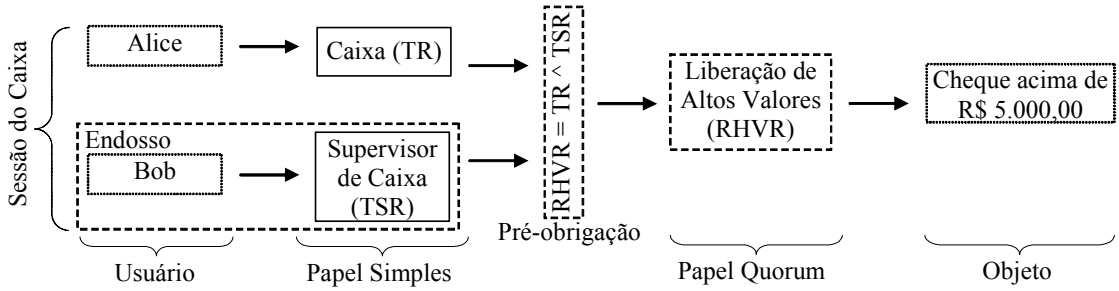

Figura 3. Exemplo de ativação de um papel quorum

No caso apresentado na Figura 3, o supervisor de caixas (Bob) ativaria o seu papel na sessão do caixa (Alice) para endossar o pagamento do cheque com valor acima de R$ 5.000,00.

A Figura 4 ilustra a dinâmica de estabelecimento de uma sessão RBAC, considerando a extensão proposta. O usuário inicializa a sessão (evento S1.0) através da interface de usuário, acionando o controlador RBAC.

Após recuperar os papéis associados ao usuário o controlador RBAC mostra o conjunto de papéis disponíveis para ativação (evento S1.7). No evento S2.4 da Figura 4, é verificado se será ativado um papel simples ou papel quorum. No caso de ativação de papel simples, depois de verificadas as condições para ativação, a sessão é aberta (evento S2.5). Por outro lado, na ativação de um papel quorum o conjunto predefinido de principais é verificado (evento 2.5.6). Se todos os eventos precedendo o evento 2.5.7 são executados com sucesso a sessão é aberta. O usuário é informado do sucesso ou falha na abertura da sessão no evento S2.7.

Como a proposta suporta cenários onde os atributos podem ser modificados dinamicamente, ou seja, a qualquer momento um atributo relacionado a um usuário ou a

um objeto pode ser modificado (mutabilidade de atributos) as obrigações devem ser continuamente verificadas (evento S3.0, Figura 4).

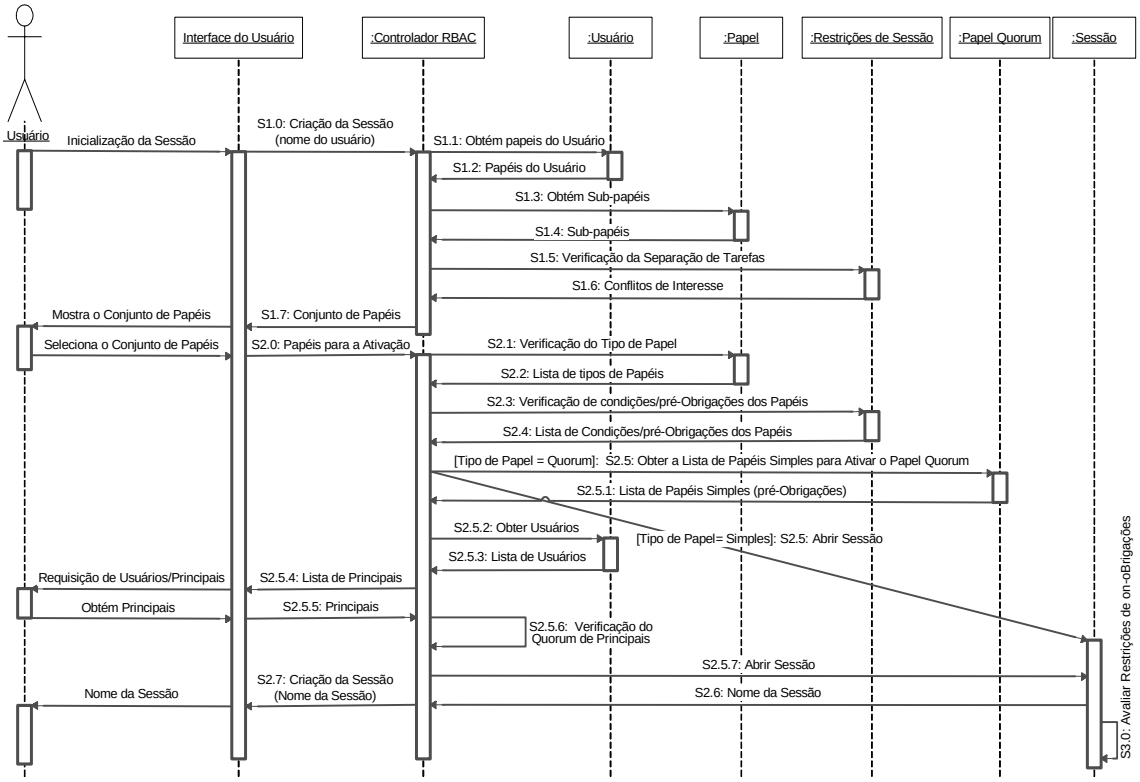

Figura 4. Extensões ao modelo de restrições RBAC aplicadas a uma sessão

A ativação do papel quorum pode ser limitada a um intervalo de tempo. Neste caso, quando o tempo expira o controlador RBAC desativa o papel quorum e reajusta as permissões do usuário de acordo com o estado anterior à ativação do papel quorum. Restrições de tempo de uso são definidas por on-B (on-obrigações) no modelo proposto.

A seguir será descrito um cenário simples que ilustra o uso das extensões propostas. Observa-se que o cenário de aplicação poderia ser outro qualquer onde haveria necessidade das características de mutabilidade de atributo apresentadas nesta proposta.

4. Cenário: Administração de Rede

Redes e sistemas de gerenciamento são tarefas críticas para muitas organizações. Em grandes organizações estas funções são realizadas por diferentes papéis. No cenário proposto são considerados os seguintes papéis simples: R1 (*guest*), R2 (*system operator*), R3 (*system administrator*) e R4 (*network administrator*).

A tabela 1 mostra as principais atribuições de cada papel, citado anteriormente, e de dois papéis com atribuições especiais (QR1 e QR2). A descrição das principais atribuições de cada papel dá uma idéia das políticas de gerenciamento da rede.

Com base no cenário da Figura 5, considere que o administrador de rede (Eve) por algum motivo está indisponível (ausente) temporariamente e que ocorreu um problema no roteador principal da organização, desabilitando o acesso à Internet. Para restaurar o acesso o mais rápido possível, outro operador que tenha conhecimentos sobre o roteador deve ser autorizado temporariamente a realizar a tarefa de manutenção.

Tabela 1 – Alguns papéis da organização

Papel	Descrição das principais atribuições do papel
R1	Guest: Papel com direitos de acesso muito restritos de uso do sistema
R2	System Operator: Papel com direitos restritos de administração do sistema
R3	System Administrator: Papel com direitos irrestritos de administração dos recursos do sistema
R4	Network Administrator: Papel com direitos irrestritos de administração dos recursos de rede
QR1	Papel (quorum) com direitos de visualização de configurações e efetivação de testes de equipamentos
QR2	Papel (quorum) com direitos irrestritos de administração dos recursos de rede

Várias soluções podem ser consideradas para minimizar ou evitar esta situação difiícil. Entre as soluções possíveis, poderia ser autorizado um técnico externo (isto é, um técnico da empresa fabricante do equipamento) para fazer a manutenção do equipamento de rede. Esta solução levaria a outro problema: como criar uma conta de administrador temporária para um principal externo a organização sem violar a política da organização?

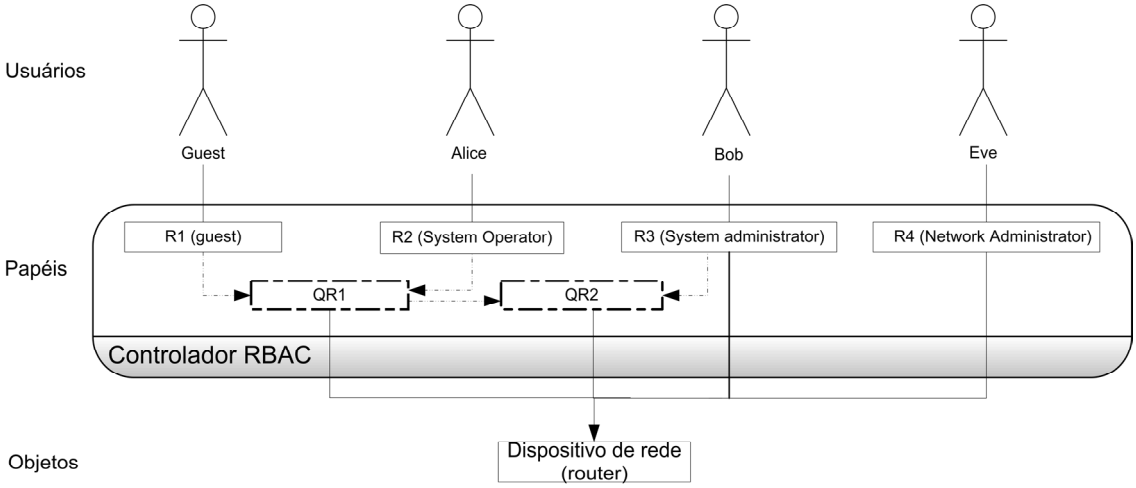

Figura 5. Mecanismo de ativação de papéis quorum para o cenário proposto

Assumiu-se que a solução escolhida foi autorizar o técnico do suporte designado pelo vendedor do equipamento a fazer a manutenção. Para conseguir acesso ao roteador principal (BGP) da organização (empresa) o técnico deve receber credenciais de autenticação (como a senha, por exemplo) para um usuário visitante (*guest*). Por padrão, um usuário visitante não possui qualquer privilégio para acessar a roteador. O papel quorum QR1 associado ao usuário *guest* pode prover tais privilégios, mas este requer o endosso do papel R2 para ser ativado.

Quando Alice ativar o papel de operador de sistema (R2) na sessão de *guest* o controlador RBAC ativará automaticamente o papel QR1, o usuário *guest* (visitante)

receberá privilégios (limitados) permitindo-lhe visualizar as configurações do roteador e fazer testes básicos no equipamento. Entretanto, *guest* ainda não obteve privilégios administrativos suficientes para modificar as configurações do roteador, ou seja, não está autorizado a executar um comando de habilitação (*enable*) no roteador, por exemplo.

Supondo que depois de verificada a situação corrente do roteador o técnico julgue necessário modificar a configuração do mesmo. Neste caso, o papel QR2 precisa ser ativado. A *on-obrigação* para a ativação do papel quorum QR2 define que é necessário o endosso do operador (R2) e do administrador do sistema (R3). Os privilégios do papel quorum QR2 são concedidos através da ativação dos papéis R2 e R3 na sessão do usuário *guest*, permitindo assim a tarefa de manutenção.

A próxima seção apresenta o protótipo e os aspectos da implementação da proposta.

5. Aspectos da Implementação

O protótipo foi efetivado com base na implementação RBAC/Web do NIST [Ferraiolo e outros, 1999], uma aplicação intranet, onde o RBAC é utilizado como esquema de autorização para controlar o acesso às páginas de um servidor HTTP. No *framework* RBAC/Web, usuários correspondem a logins no servidor. Transações HTTP que podem ser executadas por usuários (através dos seus papéis) nas páginas *html* representam as permissões RBAC. Como prova de conceito o protótipo implementa o cenário descrito na seção 4 (Figura 5). O controle de acesso do RBAC/Web foi desenvolvido numa API (disponibilizada pelo NIST em ´C´ e Perl) e os códigos são de domínio público.

Para dar suporte aos papéis quorum no protótipo foi adicionado um atributo denominado *Access Level* (AL) a cada papel RBAC. Este atributo armazena um valor numérico diferente para cada papel no sistema. O valor numérico pode representar num único atributo uma coleção de papéis simples requeridos na ativação do papel quorum.

Para evitar interpretações equivocadas de AL foram utilizados valores numéricos resultantes da expressão 2^n para os papéis simples, com n = 0,1,2,3 etc.; n representa o número de papéis simples do sistema. Para os papéis quorum o AL é resultado da expressão:

$$AL = \sum_{x=0}^{(n-1),k=0,1} 0^k \cdot 2^x$$

;onde k pode ser: k= 0 (default) ou k = 1 (se o papel simples com AL = 2^x é requerido para o endosso).

Considerando que no cenário da Figura 6 os papéis simples tenham os seguintes ALs: R1 = 64, R2 = 32, R3 = 16, então o papel quorum QR1 = 96 (resulta dos ALs de R1 e R2) e QR2 = 112 (resulta dos ALs de R1, R2 e R3). A Figura 6 mostra a obtenção dos ALs para o papel quorum a partir do cenário da Figura 5.

A efetivação do protótipo foi alcançada fazendo-se adaptações nas classes da implementação RBAC/Web. Note que para suportar nossa proposta alguns novos métodos precisaram ser desenvolvidos.

O método privado *writeAccessLevel* na classe *Role* é executado no momento da criação do papel pelo método *addRole*, e grava o papel e seu nível de acesso no arquivo *AL_role* para registrar o relacionamento entre ambos. Na exclusão de um papel (através do método *DeleteRole*) seu nível de acesso é marcado como inativo ao invés de ser apagado; o AL é preservado pelo método *rmAccessLevel* para fins de auditoria e nunca é reutilizado no sistema. O método *CreateSession* da classe *Session* invoca os métodos privados: *create_login_choices* (cria o conjunto de papéis que o usuário pode ativar, pois não possuem separação dinâmica de tarefas) e *writeARS* (grava os papéis selecionados pelo usuário no arquivo *user_name.active_roles*). Antes da execução do método *writeARS* é invocado o método *checkAccessLevel* – que foi incluído para verificar se existe algum papel quorum no conjunto de papéis que o usuário está tentando ativar. Em caso afirmativo – há papel quorum a ativar, é verificada a existência do quorum de principais requeridos para a ativação.

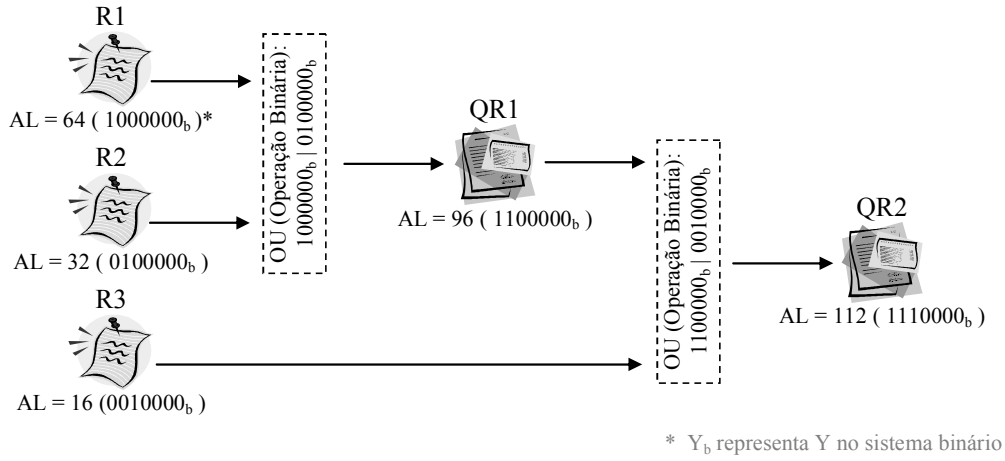

Figura 6. *Access Level* (AL) como mecanismo de ativação de papel quorum

Quando um papel quorum é ativado o método *writeAccessLog* grava informações adicionais ao log, como: data/hora da ativação, nome do papel quorum ativado e principais/papéis que endossaram a ativação. Todas as ações envolvendo a ativação de papéis quorum geram registros no arquivo de log, que são acrescidos aos dados comumente encontrados neste tipo de registro, pois todos os dados dos participantes da sessão também serão registrados.

No protótipo uma página *html* é oferecida para o *login* no sistema de administração do roteador. Quando o usuário *guest* faz *login*, só pode visualizar algumas informações do dispositivo (roteador). Para visualizar as configurações da interface do servidor e do roteador e fazer testes sem a carga da rede, por exemplo, o controlador RBAC solicitada o *login* do *system operator* (como endosso) para ativar o papel quorum QR1 e permitir tal operação. O mesmo acontece quando o usuário *guest* tenta alterar a configuração do roteador, o *login* do *operator* e do *administrator* são solicitados. Se o *login* e a senha estão corretos o usuário *guest* consegue fazer as alterações desejadas, mas o *operator* e o *administrator* não recebem tais privilégios, ambos só endossam a ativação do papel QR2 e continuam seus trabalhos como usuários simples em suas respectivas sessões RBAC.

O período de validade de um papel quorum ativo numa sessão é uma restrição opcional oferecida na tela de endosso no protótipo implementado.

6. Trabalhos Relacionados

Esta seção discute alguns trabalhos relacionados que propõem alternativas ao modelo de restrições NIST RBAC e uma implementação do modelo $UCON_{ABC}$ para computação colaborativa.

O framework proposto por [Strembeck e Neumann, 2004] é baseado em restrições que não são parte do *core* do modelo RBAC, mas que são levadas em consideração na tomada de decisão do serviço de controle de acesso. Neste caso, as restrições são denominadas restrições de contexto (*context constraints*) e impõem dinamicamente restrições de autorização a cada contexto de aplicação (em computação ubíqua, por exemplo).

A proposta apresentada por [Strembeck e Neumann, 2004] facilita a expressão de separação de tarefas de uma maneira não abordada pelo modelo RBAC original. Entretanto, a proposta de utilização da SoD segue o comportamento original do RBAC, separando uma tarefa em subtarefas, organizadas de maneira que impõem a participação de diferentes principais para execução de cada subtarefa.

Em [Zhang e Nakae, 2006] é utilizado o $UCON_{ABC}$ para organizações virtuais no contexto de uma infra-estrutura segura de grid. Na proposta obrigações não são abordadas; somente autorizações e condições são consideradas. É aplicado o modelo PCIM *outsourcing* (RFC3460), baseado nas entidades PEP (*Policy Enforcement Point*) e PDP (*Policy Decision Point*) para implementar $UCON_{ABC}$. A contínua avaliação dos atributos mutáveis é feita pelo PDP, que atualiza a condição do sujeito/objeto de acordo com as políticas vigentes. O protótipo é implementado com um módulo do Apache WebDAV (http://webdav.org/mod_dav).

A proposta apresentada em [Zhang e Nakae, 2006] na verdade implementa controle de acesso clássico em ambiente dinâmico considerando continuidade e mutabilidade de atributos.

Em nossa proposta o endosso para ativar um papel não é avaliado de acordo com os atributos de controle de acesso mutável para evitar o surgimento de possíveis conflitos na alteração dos direitos de um papel em tempo de execução. Estas alterações podem levar a conflito de interesses devido a alteração de atributos depois do RBAC ter avaliado a separação dinâmica de tarefas (DSD). Ao invés disto, nossa proposta aplica os princípios de continuidade e mutabilidade aos papéis RBAC estritamente no contexto de controle de uso dos papéis. Além disto, a mutabilidade de atributos deve permitir ao controlador RBAC a ativação ou desativação automática de papéis e não a mudança dos atributos de autorização como mencionado anteriormente.

7. Conclusão

Este trabalho apresentou extensões ao modelo de restrições de consenso NIST RBAC. O objetivo foi apresentar uma abordagem capaz de acomodar os requisitos de um ambiente no qual a separação de tarefas não pode ser executada pela divisão de uma tarefa em subtarefas executadas por diferentes papéis. Em nossa poroposta, um quorum de papéis

distintos é requerido para endossar a execução de uma tarefa que um papel sozinho não conseguiria realizar. O modelo de restrições do RBAC não tem suporte para este tipo de autorização.

O RBAC não suporta a mutabilidade de atributos do sujeito. Entretanto, através da utilização de um quorum de papéis foi possível obter um comportamento similar ao modelo $UCON_{ABC}$. Visto que não existe qualquer implementação considerando as obrigações do modelo $UCON_{ABC}$, a proposta pode ser aplicada para avaliar aspectos relativos à mutabilidade e continuidade em sistemas reais.

Não há riscos de segurança com relação a ativação não autorizada do papel quorum, pois a ativação e controle do mesmo é feita em nível de controlador RBAC, e não em nível de usuário.

No mecanismo de controle de acesso não há entre os níveis de acesso (AL) qualquer hierarquia entre os papéis simples na composição de papéis quorum – esta característica é desejada para se conseguir a escalabilidade do sistema. Como a combinação de ALs gera um novo AL, um papel quorum deve ser explicitamente vinculado a um AL, evitando interpretações equivocadas na combinação de atributos usados na criação das políticas, por exemplo.

O protótipo mostrou que a estratégia proposta pode ser facilmente implementada com poucas modificações no framework RBAC/Web. Para ilustrar a aplicabilidade da extensão proposta, um cenário típico de manutenção de redes foi avaliado. Neste cenário, um usuário *guest* foi temporariamente autorizado por empregados da corporação/organização a realizar uma tarefa de manutenção, sem violar a política de segurança corporativa e sem modificar as associações dos papéis.

De acordo com a estratégia de papéis quorum, utilizando obrigações, um usuário recebe informações de quais papéis são necessários ativar para realizar uma tarefa que naquele momento não possui privilégios para executar. Esta abordagem é inovadora em relação ao comportamento dos mecanismos tradicionais de controle de aceso, que neste caso somente negariam o acesso e não informariam ao usuário o que precisaria fazer para conseguir o acesso.

Atualmente, o mecanismo de autenticação do protótipo é baseado no esquema usuário/senha. No futuro, este mecanismo deve ser melhorado ou substituído por um esquema mais flexível, talvez baseado em credenciais, por exemplo.

Referências

Brewer, D., Nash, M., (1989) “The Chinese wall security policy”, In Proceedings of the Symposium on Security and Privacy, IEEE Press.

Ferraiolo, D., Barkley, J., Kuhn, R., (1999) “A Role Based Access Control Model and Reference Implementation within a Corporate Intranet”, In Proceedings of NIST. Acessado em Janeiro, 2007, http://hissa.nist.gov/rbac/RBACdist/rbac_v1.1_dist.tar.

Ferraiolo, D., Kuhn, R., (1992) “Role-Based Access Control”, In Proceedings of NIST - NCSC National Computer Security Conference.

Ferraiolo, D., Sandhu, R., Gavrila, S., Kuhn,D.R., Chandramouli, R., (2001) “A Proposed Standard for Role Based Access Control”, In ACM Transactions on Information and System Security, vol. 4, no. 3.

Jaehong, P. Xinwen, Z. Sandhu, R., (2004) “Attribute Mutability in usage control”, In proceeding of DBSec’2004, pg. 15-19.

Nyachama, M., Osborn, S., (1999) “The Role Graph Model and Conflict of Interest”, In ACM TISSEC, vol. 2, no.1.

Park, J., Sandhu, R., (2004) “The $UCON_{ABC}$ usage control model”, In ACM Transactions on Information and System Security, Vol. 7, Issue 1.

Saltzer, J.H., Schroeder, M.D., (1975) “The Protection of information in computer systems”, In proceedings of IEEE, vol. 63, no. 9, pp. 1278-1308.

Sandhu, R., (1998) “Role-Based Access Control”, In Advances in Computers. Academic Press, vol. 46.

Sandhu, R., Ferraiolo, D., Kuhn, R., (2000) “The NIST Model for Role-Based Access Control: Towards A Unified Standard”, In Proceedings of ACM Workshop on Role-Based Access Control, ACM Press.

Shoup, V., (2000) “Practical threshold signatures”, In Proceedings of Eurocrypt.

Simon, R., Zurko, M. E., (1997) “Separation of Duty in Role-based Environments”, In Proceedings of the 10th Computer Security Foundations Workshop.

Strembeck, M., Neumann, G., (2004) “An Integrated Approach to Engineer and Enforce Context Constrain in RBAC Enviroments”, In ACM transaction on Information and System Security, Vol. 7, no. 3.

Zhang, X., Nakae, M. Covington, M., Sandhu, R., (2006) “A usage-based authorization framework for collaborative computing systems”, In Proceedings of the eleventh ACM symposium on Access control models and technologies, pag. 180 – 189.